Lab #5
Differential expression

In this lab, we will be conducting a two-sample test for each gene/probe on the array to identify differentially expressed genes/probes between ketogenic rats and control diet rats. This small data set was run on the rat RAE230A Affymetrix array. The objective of the study was to determine differences in mRNA levels between brain hippocampi of animals fed a ketogenic diet (KD) and animals fed a control diet. "KD is an anticonvulsant treatment used to manage medically intractable epilepsies", so differences between the 2 groups of rats can provide biological insight into the genes that are regulated due to the treatment.

We are going to identify those genes/probes that are differentially expressed between the 2 rat diet groups and plot the results with a couple of different visual summaries.

1.) Download the GEO rat ketogenic brain data set and save as a text file.

2.) Load into R, using read.table() function and header=T/row.names=1 arguments.

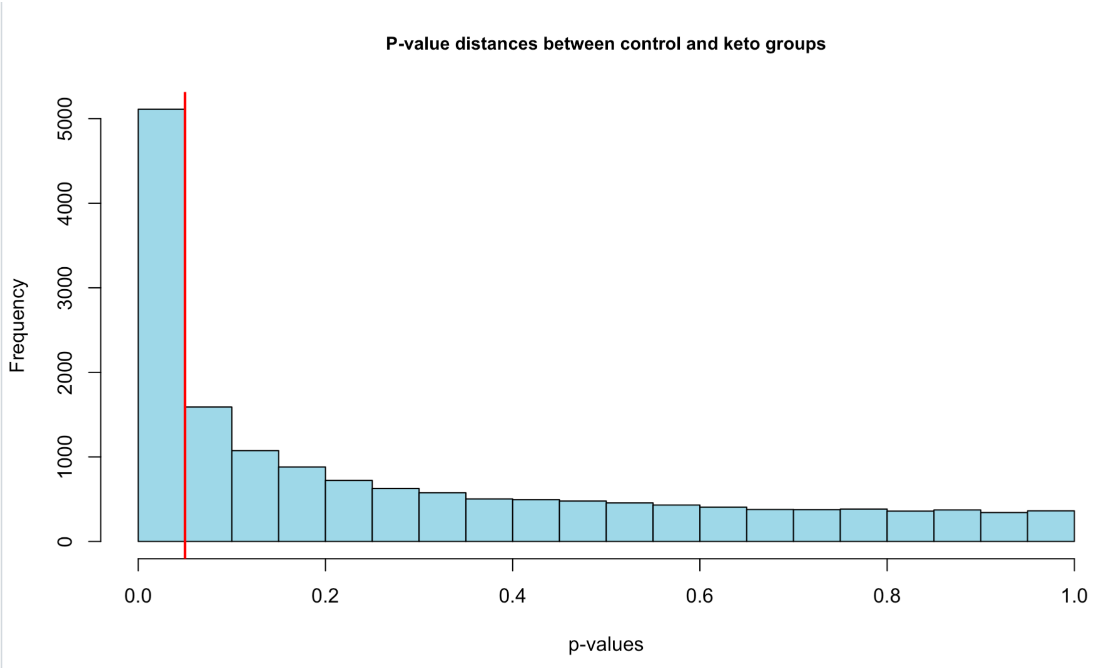
```
> lab5 <- read.table("/Users/stevendea/Desktop/JHU/Fall 2019/Gene Expression Data Analysis and Visualization/Labs/Lab 5/rat_KD.txt", header=T, row.names=1)
```

3.) First log$_2$ the data, then use the Student's t-test function in the notes to calculate the changing genes between the control diet and ketogenic diet classes. (Hint: use the names() function to determine where one class ends and the other begins).

```
> lab5.log2 <- log2(lab5)

#subscript control vs keto
> control <- colnames(lab5[1:6])
> keto <- colnames(lab5[7:11])

#function to perform a student's t-test on all of the genes looked at:
t.test.all.genes <- function(x,s1,s2) {
   x1 <- x[s1]
   x2 <- x[s2]
   x1 <- as.numeric(x1)
   x2 <- as.numeric(x2)
   t.out <- t.test(x1,x2, alternative="two.sided",var.equal=T)
   out <- as.numeric(t.out$p.value)
   return(out)
}

#student's t-test
pv <- apply(lab5, 1, t.test.all.genes, s1=control, s2=keto)
```

4.) Plot a histogram of the p-values and report how many probesets have a p<.05 and p<.01. Then divide an alpha of 0.05 by the total number of probesets and report how many probesets have a p-value less than this value. This is a very conservative p-value thresholding method to account for multiple testing called the Bonferroni correction that we will discuss in upcoming lectures.

```
#graph p-values on a histogram
> hist(pv, col="lightblue", xlab="p-values", main="P-value distances between control and keto groups", cex.main=0.9)
> abline(v=.05,col=2,lwd=2)
```



```
#Get all values that are less than p of .05
> pv.05 <- pv < 0.05
> sum(pv.05)
[1] 5112 p-values <0.05

#values of p < 01
> pv.01 <- pv < 0.01
> sum(pv.01)
[1] 2453
#0.05/15923 probesets
threshold <- 0.05/15923
[1] 3.140112e-06

#how many p-values below this number
> pv.threshold <- pv < threshold
> sum(pv.threshold)
```

```
[1] 11 p-values below the threshold.
```

5.) Next calculate the mean for each gene, and calculate the fold change between the groups (control vs. ketogenic diet). Remember that you are on a log$_2$ scale.

```
#calculate mean for each class/gene of controls
> control.m <- apply(lab5.log2[,control], 1, mean, na.rm=T)
> keto.m <- apply(lab5.log2[,keto], 1, mean, na.rm=T)

#get the log2 fold change between control and keto
> fold <- control.m - keto.m
```

6.) What is the maximum and minimum fold change value, please report on the linear scale? Now report the probesets with a p-value less than the Bonferroni threshold you used in question 4 **and** |fold change|>2. Remember that you are on a log$_2$ scale for your fold change and I am looking for a linear |fold| of 2.

```
#fold change for linear scale
> control.m.linear <- apply(lab5[,control], 1, mean, na.rm=T)
> keto.m.linear <- apply(lab5[,keto], 1, mean, na.rm=T)
> fold.linear <- control.m.linear-keto.m.linear
> max(fold.linear)
[1] 1588.693
> min(fold.linear)
[1] -1315.219

#find which are true for both Bonferroni threshold and |fold change| > 2
#numbers underneath probeset indicate the index they are located
> pv.threshold.fold <- (pv < threshold) & (abs(fold.linear) > 2)
> which(pv.threshold.fold)
1367553_x_at   1368071_at   1370239_at 1370240_x_at
         578         1289         4367         4368
  1370355_at 1371102_x_at 1388608_x_at   1373040_at
        4519         5562         6943         8329
  1374641_at   1390092_at   1376005_at
       10595        12290        12533
```

7.) Go to NetAffx or another database source if you like and identify gene information for the probesets that came up in #6. What is the general biological function that associates with these probesets?

1367553_x_at - hemoglobin, glutathione metabolic process

1368071_at - mithocondrial amidoxime, metabolic process

1370239_at – hemoglobin, in utero embryonic development

1370240_x_at – hemoglobin, in utero embryonic development

1370355_at – stearoyl-coenzyme A desaturase 1 – lipid metabolic process

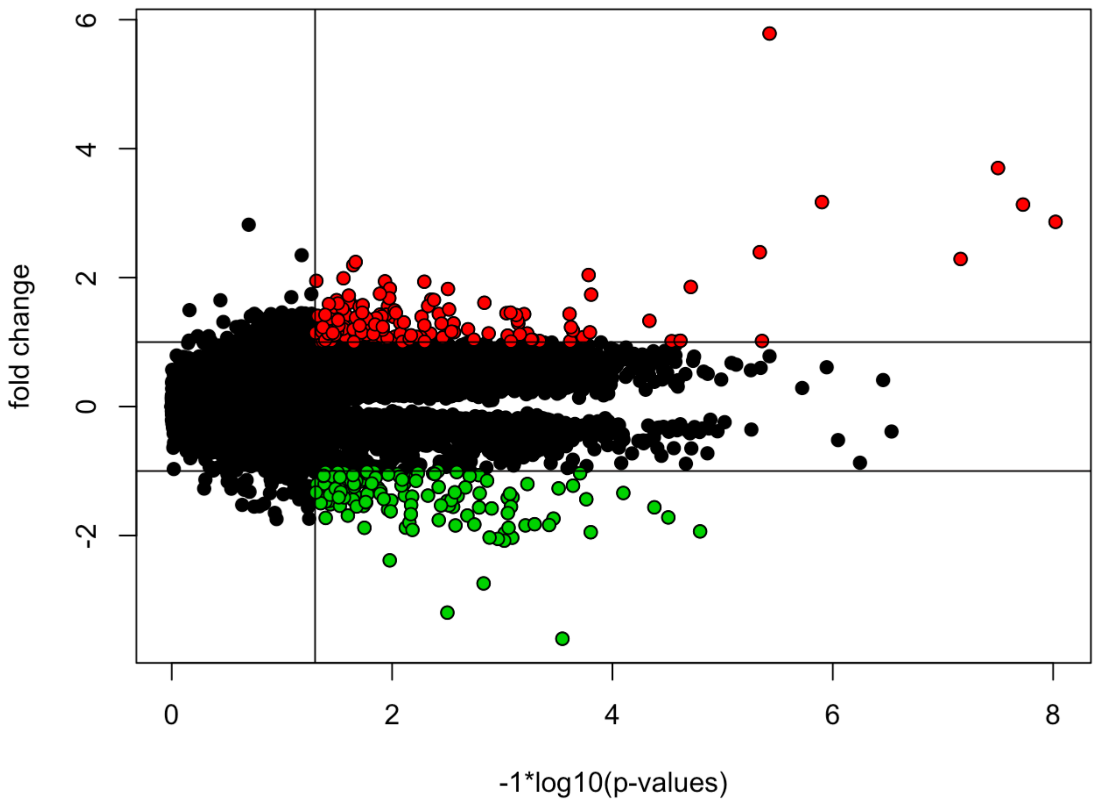
1371102_x_at – hemoglobin, oxygen transport

1388608_x_at - hemoglobin alpha 1, 2, in utero embryonic development
1373040_at – eukaryotic translation initiation factor 3 subunit F, formation of translation preinitiation complex
1374641_at - lemur tyrosine kinase 2, receptor recycling
1390092_at - No biological process found in affymetrix
1376005_at – No biological process found in affymetrix

8.) Transform the p-value (-1*log10(p-value)) and create a volcano plot with the p-value and fold change vectors (see the lecture notes). Make sure to use a log$_{10}$ transformation for the p-value and a log$_2$ (R function log2()) transformation for the fold change. Draw the horizontal lines at fold values of 2 and -2 (log$_2$ value=1) and the vertical p-value threshold line at p=.05 (remember that it is transformed in the plot).

```
#transform p-value and fold change
> transformed.pv <- (-1*(log10(pv)))
> control.m <- apply(lab5.log2[,control], 1, mean, na.rm=T)
> keto.m <- apply(lab5.log2[,keto], 1, mean, na.rm=T)
> fold <- control.m - keto.m

#volcano plot with horizontal lines at fold 2, -2 & vertical p=.05

#plot points
> plot(range(transformed.pv), range(fold), type='n', xlab='-1*log10(p-values)',
ylab='fold change', main="Volcano Plot of Control vs. Keto groups")
> points(transformed.pv, fold, col='black', pch=21, bg=1)
> points(transformed.pv[(transformed.pv> -
log10(.05)&fold>log2(2))],fold[(transformed.pv> -
log10(.05)&fold>log2(2))],col=1,bg=2,pch=21)
> points(transformed.pv[(transformed.pv> -log10(.05)&fold< -
log2(2))],fold[(transformed.pv> -log10(.05)&fold< -log2(2))],col=1,bg=3,pch=21)

#plot lines
> abline(v= -log10(.05))
> abline(h= -log2(2))
> abline(h=log2(2))
```

Volcano Plot of Control vs. Keto groups


fold change
-1*log10(p-values)